AMENDMENTS TO THE CLAIMS

1. (Currently Amended) [[A]] In a computer system comprising a microprocessor, a method of providing partial speculative operation in lieu of suspending speculation, said method comprising:

operating in a first mode of speculative operation, said first mode permitting speculation of a first set of speculative operations;

experiencing an event during said operating;

in response to said event, restoring a state of said microprocessor to a state committed to memory prior to said event;

subsequent to said restoring, suspending a non-null first subset of said first set of speculative operations, wherein speculative operations in said first subset are not permitted during said suspending; and

exiting said first mode and entering a second mode of speculative operation in response to said event, said second mode permitting speculation of a non-null second subset of said first set, ~~wherein said second subset comprises speculative operations not in said first subset, wherein said first set of speculative operations comprises operations that involve memory that is private to a microprocessor,~~ wherein said first subset comprises input/output (I/O) writes, main memory reads, and main memory writes, wherein said first mode permits speculation for handling non-architectural faults, wherein said second subset comprises operations that involve memory that is private to said [[a]] microprocessor, and wherein said second mode permits speculation for handling architectural faults.

2-3. (Canceled).

4. (Currently Amended) The method of Claim 1 wherein said second subset comprises speculative operations that are invisible external to said [[a]] microprocessor.

5. (Previously Presented) The method of Claim 1 wherein said event is selected from the group consisting of a fault, and a direct memory access request.

6. (Original) The method of Claim 1 further comprising suspending speculative operation in response to a second event.

7. (Original) The method of Claim 1 further comprising returning to said first mode after said event is handled.

8. (Original) The method of Claim 1 further comprising:

counting the number of instructions executed in said first mode prior to said event; and

returning to said first mode upon executing the same number of instructions after entering said second mode.

9. (Currently Amended) The method of Claim 1 wherein said microprocessor comprises ~~implemented using a microprocessor comprising~~ a combination of translation software and host hardware, said translation software running directly on said host hardware, said translation software for interpreting and translating a sequence of non-native instructions into a sequence of native instructions, wherein said interpreting is permitted during said second mode.

10. (Currently Amended) A method providing partial speculative operation, said method comprising:

executing forward from a speculation boundary representing a memory state of a microprocessor, said executing according to a full speculation mode that permits a set of speculative operations;

experiencing an event during said executing;

rolling back to said speculation boundary and restoring said memory state in response to said event;

subsequent to said rolling back, suspending a non-null first subset of said set of speculative operations, wherein said first subset does not include all of said speculative operations and wherein speculative operations in said first subset are not permitted during said suspending; and

subsequent to said suspending, executing forward from said speculation boundary according to a partial speculation mode that permits a non-null second subset of said set of speculative operations, ~~wherein said second subset comprises speculative operations not in said first subset and~~ wherein said partial speculation mode is used in lieu of suspending said set of speculative operations in entirety, ~~wherein said first set of speculative operations comprises operations that involve memory that is private to a microprocessor,~~ wherein said first subset comprises input/output (I/O) writes, main memory reads, and main memory writes, wherein said full speculation mode permits handling non-architectural faults, wherein said second subset comprises operations that involve memory that is private to said [[a]] microprocessor, and wherein said partial speculation mode permits handling architectural faults.

11-12. (Canceled).

13. (Currently Amended) The method of Claim 10 wherein said second subset of speculative operations comprises speculative memory operations that are invisible external to said [[a]] microprocessor.

14. (Previously Presented) The method of Claim 10 wherein said event is selected from the group consisting of a fault, and a direct memory access request.

15. (Original) The method of Claim 10 further comprising:

detecting a second event during operation in said partial speculation mode; and

suspending speculative operation in response to said second event.

16. (Original) The method of Claim 10 further comprising:

handling said event; and

returning to said full speculation mode after said event is handled.

17. (Original) The method of Claim 10 further comprising:

counting the number of instructions executed in said full speculation mode prior to said event;

executing the same number of instructions after entering said partial speculation mode; and

returning to said full speculation mode after executing said same number of instructions.

18. (Currently Amended) The method of Claim 10 wherein said microprocessor comprises ~~implemented using a microprocessor comprising~~ a combination of translation software and host hardware, said translation software running directly on said host hardware, said translation software for interpreting and translating a sequence of non-native instructions into a sequence of native instructions, wherein said interpreting is permitted during said partial speculation mode.

19. (Currently Amended) A computer system comprising:

a main memory; and

a microprocessor coupled to said main memory;

wherein said computer system implements a first mode of speculative operation, a second mode of partial speculative operation, and a third mode in which said speculative operations are suspended in entirety, wherein operation in said first mode permits speculation of a set of speculative operations up to and beyond a commit point at which a state of said microprocessor is committed to memory and wherein operation in said second mode begins after a rollback operation that restores said state, and wherein in said second mode a non-null first subset of said set of speculative operations are suspended leaving enabled a non-null second subset of said set of speculative operations, wherein said first mode permits speculative operations comprising operations that involve memory that is private to said microprocessor, input/output (I/O) writes, main memory reads, and main memory writes, wherein said first mode permits speculation for non-architectural faults, wherein said second mode permits speculative operations comprising operations that is private to said microprocessor, and wherein said second mode permits speculation for handling architectural faults.

20-21. (Canceled).

22. (Original) The computer system of Claim 19 wherein said second mode permits speculative operations that are invisible external to said microprocessor.

23. (Previously Presented) The computer system of Claim 19 wherein a transition from said first mode to said second mode occurs in response to an event that is selected from the group consisting of a fault, and a direct memory access request.

24. (Original) The computer system of Claim 23 wherein a transition back to said first mode occurs after said event is handled.

25. (Original) The computer system of Claim 23 wherein the number of instructions executed in said first mode prior to said event are counted, wherein a transition back to said first mode occurs after the same number instructions are executed in said second mode.

26. (Original) The computer system of Claim 19 wherein said microprocessor is a microprocessor comprising a combination of translation software and host hardware, said translation software running directly on said host hardware, said translation software for interpreting and translating a sequence of non-native instructions into a sequence of native instructions, wherein said interpreting is permitted during said second mode.